Real-time Operating Systems and Systems Programming

Bit-fields & Variable arguments
Lecture 11

ARVUTITEADUSE INSTITUUT
DEPARTMENT OF COMPUTER SCIENCE

Bit fields

- An alternative to flag variables. It is possible to "pack" values into a structure using : notation to signify the amount of bits allocated. Assignment and use like a regular structure.

```
struct {
   unsigned int is_keyword: 1;
   unsigned int is_extern: 1;
   unsigned int is_static: 1;
} flags;

flags.is_static = 1;
flags.is_keyword = 0;
```

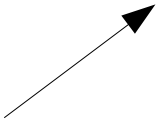
Bit-fields (2)

- Bit-fields are declared as unsigned integers to avoid sign problems.
- For use in expressions they are cast into an integer automatically.
- Internal implementation depends on architecture.

Variable arguments

- To declare functions with variable arguments, just type ... like you would not care about what goes there...

```
void foo( int arg1, int arg2, ...) {
```



Ellipsis here...

- How to get to them?

Getting those other arguments

Whiteboard fun

Getting the other arguments using a macro (easier method)

- #include <stdarg.h>
- Macros:

```
void va_start(va_list ap, last);
type va_arg(va_list ap, type);
void *va_end(va_list ap);
```

- Possible implementation

```
typedef char *va_list;
#define va_start(ap, v)  ((void) (ap = (va_list) &v + sizeof(v)))
#define va_arg(ap,type)   (*((type *)(ap))++)
#define va_end(ap)  ((void) (ap = 0))
```

More info: 'man stdarg' or 'man vararg'

Var args (example)

```
#include <stdarg.h>
#define MAXARGS     31
void f1(int n_ptrs, ...) {
    va_list ap;
    char *array[MAXARGS];
    int ptr_no = 0;

    if (n_ptrs > MAXARGS)
        n_ptrs = MAXARGS;
    va_start(ap, n_ptrs);
    while (ptr_no < n_ptrs)
         array[ptr_no++] = va_arg(ap, char*);
    va_end(ap);
    f2(n_ptrs, array);
}
```